UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/789,311	02/27/2004	Sheueling Chang Shantz	6000-31500	9201

58467 7590 09/27/2011

MHKKG/Oracle (Sun)
P.O. BOX 398
AUSTIN, TX 78767

EXAMINER	
JOHNSON, CARLTON	

ART UNIT	PAPER NUMBER
2436	

NOTIFICATION DATE	DELIVERY MODE
09/27/2011	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patent_docketing@intprop.com
ptomhkkg@gmail.com

Office Action Summary	Application No.	Applicant(s)	
	10/789,311	SHANTZ ET AL.	
	Examiner	Art Unit	
	CARLTON JOHNSON	2436	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 July 2011*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

5)☒ Claim(s) *1-67* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *1-67* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

12)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to application amendments filed on 7-1-2011.

2. Claims **1 - 67** are pending. Claims **1, 21, 38, 53, 66, 67** are independent. This application was filed on 2-27-2004.

Response to Arguments

3. Applicant's arguments have been fully considered and they were partially persuasive, therefore new grounds of rejection have been entered. .

A. Applicant argues on page 16 of Remarks *that Takahashi fails to disclose a first arithmetic circuit comprising a first plurality of arithmetic structures feeding back high order bits of a previously executed single arithmetic instruction of a processor instruction set in the public-key cryptography application, generated by the first arithmetic circuit, to a second arithmetic circuit comprising a second plurality of arithmetic structures.*

The Examiner disagrees. Takahashi discloses a processor capability of performing multiple pipeline operations from a single instruction, which is part of the processor architecture (or processor instruction set). And, Takahashi discloses a system capable of performing cryptographic type operations such as encryption and decryption of data. In addition, Takahashi discloses the capability to generate a partial result (reduced number of resulting bits) from previous operations within a sequence of processing

operations. (see Takahashi paragraph [0009], lines 1-3: multi-function processor architecture capable of performing mathematics operations (implies arithmetic operations are part of system architecture or instruction set); paragraph [0017], lines 1-8: system for encrypting and decrypting data; paragraph [0017], lines 1-19: pipeline storage processing stage iteratively computing a running partial product using one or more received operands a predetermined number of times; post processing stage to receive final partial product and compute result; paragraph [0044], lines 1-14: receive bit stored in the current highest order position; first carry-save processor (a0 position); second carry-save processor (a1 position); third carry-save processor (a2); ...)

Takahashi discloses the generation of a partial result. And, Chen discloses the capability for a partial result to consist of a high order portion of the result. (see Chen paragraph [0017], lines 14-19: hardware utilized to perform addition and multiplication; paragraph [0100], lines 12-18: output is high order bits, storing high order bits from adder; paragraph [0274], lines 16-19: carry feedback out of high order position)

And, Yeh discloses the capability for the transfer of an operand between two separate arithmetic instructions such as feeding back bits of a previously executed single arithmetic instruction (or an operand transferred between two separate instructions). (see Yeh col 5, ll 27-33: first instruction in a dependency chain accepts input from registers; second instruction accepts results of first instruction as it input operand)

B. Applicant argues on page 17 of Remarks *that there is nothing in the cited*

passages, or elsewhere in Takahashi, that describes feeding back high order bits of a previously executed single arithmetic instruction of a processor instruction set that were generated by a first arithmetic circuit, to a second arithmetic circuit.

The Examiner disagrees. Takahashi discloses a processor with the capability of performing multiple pipeline operations from a single instruction. Chen discloses the capability for a partial result consisting of the high order portion of the result. (see Chen paragraph [0017], lines 14-19: hardware utilized to perform addition and multiplication; paragraph [0100], lines 12-18: output is high order bits, storing high order bits from adder; paragraph [0274], lines 16-19: carry feedback out of high order position) And, Yeh discloses the capability for the transfer of an operand between two arithmetic instructions such as feeding back bits of a previously executed single arithmetic instruction (or an operand transferred between two separate instructions). (see Yeh col 5, ll 27-33: first instruction in a dependency chain accepts input from registers; second instruction accepts results of first instruction as it input operand)

C. Applicant argues on page 17 of Remarks *that Takahashi does not describe any instructions of the processor's instruction set, much less any type of feedback that occurs between two different instructions (i.e. two instances of any of the instructions of the processor's instruction set) when they are executed.*

The Examiner disagrees. Takahashi discloses a processor capability of performing multiple pipeline operations from a single instruction. And, Yeh discloses the capability

for the transfer of an operand between two arithmetic instructions such as feeding back bits of a previously executed single arithmetic instruction (or an operand transferred between two separate instructions). (see Yeh col 5, ll 27-33: first instruction in a dependency chain accepts input from registers; second instruction accepts results of first instruction as it input operand)

D. Applicant argues on page 17 of Remarks *that Takahashi fails to disclose the second arithmetic circuit generating a first partial result of a currently executing single arithmetic instruction of the processor instruction set in the public-key cryptography application, wherein the currently executing single arithmetic instruction does not include an explicit source operand for specifying the high order bits', the first partial result representing the high order bits" summed with low order bits" of a result of a first number multiplied by a second number, the summing of the high order bits" being performed during multiplication of the first number and the second number, the summing and at least a portion of the multiplication being performed in the second arithmetic circuit.*

The Examiner disagrees. Takahashi discloses a processor capability of performing multiple pipeline operations from a single instruction, which is part of the processor architecture (or processor instruction set). And, Takahashi discloses a system capable of performing cryptographic type operation such as encryption and decryption of data. In addition, Takahashi discloses the capability to generate a partial result (reduced number of resulting bits) from previous operations within a sequence of processing

operations. (see Takahashi paragraph [0009], lines 1-3: multi-function processor architecture capable of performing mathematics operations (implies arithmetic operations are part of system architecture or instruction set); paragraph [0017], lines 1-8: system for encrypting and decrypting data; paragraph [0017], lines 1-19: pipeline storage processing stage iteratively computing a running partial product using one or more received operands a predetermined number of times; post processing stage to receive final partial product and compute result; paragraph [0044], lines 1-14: receive bit stored in the current highest order position; first carry-save processor (a0 position); second carry-save processor (a1 position); third carry-save processor (a2); ...)

Takahashi discloses the generation of a partial result. And, Chen discloses the capability for a partial result to consist of the high order portion of the result. (see Chen paragraph [0017], lines 14-19: hardware utilized to perform addition and multiplication; paragraph [0100], lines 12-18: output is high order bits, storing high order bits from adder; paragraph [0274], lines 16-19: carry feedback out of high order position)

Takahashi discloses a processor capability of performing multiple pipeline operations from a single instruction. And, Yeh discloses the capability for the transfer of an operand between two separate arithmetic instructions such as feeding back bits of a previously executed single arithmetic instruction (or an operand transferred between two separate instructions). (see Yeh col 5, ll 27-33: first instruction in a dependency chain accepts input from registers; second instruction accepts results of first instruction as it input operand)

E. Applicant argues on pages 17, 18 of Remarks *that Contrary to the Examiner's suggestion, the operands that are stored in the operand storage portion of the processor and that are received by the pipeline portion of the processor are not high order bits of a previously executed single arithmetic instruction of the processor's instruction set.*

The Examiner disagrees. Takahashi discloses the generation of a partial result. And, Chen discloses the capability for a partial result to consist of the high order portion of the result. (see Chen paragraph [0017], lines 14-19: hardware utilized to perform addition and multiplication; paragraph [0100], lines 12-18: output is high order bits, storing high order bits from adder; paragraph [0274], lines 16-19: carry feedback out of high order position)

F. Applicant argues on page 18 of Remarks *that Examiner's suggestion that this passage describes an arithmetic circuit of Takahashi receiving high order bits fed back from a previously executed single arithmetic instruction are completely unsupported in the reference itself.*

The Examiner disagrees. Takahashi discloses the generation of a partial result. And, Chen discloses the capability for a partial result to consist of the high order portion of the result. (see Chen paragraph [0017], lines 14-19: hardware utilized to perform addition and multiplication; paragraph [0100], lines 12-18: output is high order bits, storing high order bits from adder; paragraph [0274], lines 16-19: carry feedback out of

high order position) And, Yeh discloses the capability for the transfer of an operand between two arithmetic instructions such as feeding back bits of a previously executed single arithmetic instruction (or an operand transferred between two separate instructions). (see Yeh col 5, ll 27-33: first instruction in a dependency chain accepts input from registers; second instruction accepts results of first instruction as it input operand)

G. Applicant argues on page 19 of Remarks *that Applicants assert that the Examiner has not identified a single arithmetic instruction of a processor instruction set (i.e. an instruction implemented in a processor) in Takahashi whose execution causes the performance of the specific collection of operations described in Applicants' claim by the circuitry disclosed in Takahashi, or results in the relationships between successive instruction executions (i.e. between the executions of two distinct instances of single arithmetic instructions) recited in the claim.*

The Examiner disagrees. Takahashi discloses a processor capability of performing multiple pipeline operations from a single instruction. And, Yeh discloses the capability for the transfer of an operand between two arithmetic instructions such as feeding back bits of a previously executed single arithmetic instruction (or an operand) transferred between two single instructions. (see Yeh col 5, ll 27-33: first instruction in a dependency chain accepts input from registers; second instruction accepts results of first instruction as it input operand)

H. Applicant argues on page 20 of Remarks *that Takahashi does not describe implicit feedback (or an implicit relationship) between this single arithmetic instruction and a previously executed single arithmetic instruction of the processor's instruction set.*

The Examiner disagrees. Takahashi discloses a processor capability of performing multiple pipeline operations from a single instruction. And, Yeh discloses the capability for the transfer of an operand between two arithmetic instructions such as feeding back bits of a previously executed single arithmetic instruction (or an operand transferred between two single instructions). (see Yeh col 5, ll 27-33: first instruction in a dependency chain accepts input from registers; second instruction accepts results of first instruction as it input operand)

I. Applicant argues on page 20 of Remarks *that Claims 38 and 66 include limitations that are similar to those of claim 1 discussed above, and the Examiner rejected claims 38 and 66 for the same reasons as claim 1. Therefore, the arguments presented above apply with equal force to these claims, as well.*

The Examiner disagrees since independent claims 38 and 66 have similar limitations as independent claim 1 and responses to arguments against independent claim 1 answer arguments against independent claims 38 and 66.

J. Applicant argues on page 20 of Remarks *that Claims 21, 53, and 67 include limitations that are similar to those of claim 1 discussed above, and the Examiner*

rejected claims 38 and 66 for the same reasons as claim 1. Therefore, the arguments presented above apply with equal force to these claims, as well.

The Examiner disagrees since independent claims 21, 53 and 67 have similar limitations as independent claim 1 and responses to arguments against independent claim 1 answer arguments against independent claims 21, 53 and 67.

K. Applicant argues on page 21 of Remarks *that Applicants assert that nothing is Takahashi discloses these limitations of claim 21. Therefore Takahashi does not anticipate claim 21. Claims 53 and 67 include limitations similar to those of claim 21. Therefore, the arguments presented above and directed to claim 21 apply with equal force to these claims, as well.*

The Examiner disagrees since independent claims 21, 53 and 67 have similar limitations as independent claim 1 and responses to arguments against independent claim 1 answer arguments against independent claims 21, 53 and 67

L. Applicant argues on pages 21, 22 of Remarks *that The Office Action rejected claims 2, 3, 15-18, 27-29, 35 and 43-46 under 35 U.S.C. § 103(a) as being unpatentable over … Applicants respectfully traverse these rejections for at least the reasons given above in regard to the independent claims.*

The Examiner disagrees since responses to arguments against independent claims answer arguments against dependent claims.

Double Patenting

4. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., In re Berg, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); In re Goodman, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); In re Longi, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); In re Van Omum, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); In re Vogel, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and In re Thorington, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321 (c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

5. The present application 10/789311 is a continuation application of application 10/626420, having the same inventive entity. Claims 1, 21, 38, 53, 66, 67 of Application No. 10/789311 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over Claims 1, 18, 36, 43, 50, 57, 61, 64, 65 of copending application 10/626420. Although the conflicting claims are not identical, they are not patentably distinct from each other because the subject matter claimed in the instant application 10/789311 is fully disclosed in the referenced copending application 10/626420 and would be covered by any patent granted on that

copending application since the referenced copending application and the instant application are claiming common subject matter.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been completely patented.

Claims 1, 21, 38, 53, 66, 67 of Application No. 10/789311 are rejected on the grounds of nonstatutory obviousness-type double patenting as being unpatentable over claims 1, 18, 36, 43, 50, 57, 61, 64, 65 of application No. 10/626420. The whole of the current application (10/789311) are envisioned by the copending application claims and therefore is not distinct.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims **1, 4 - 10, 19, 21 - 26, 36, 38 - 42, 48, 52 - 60, 62, 66, 67** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Takahashi et al.** (US PGPUB No. **20020194237**) in view of **Chen et al.** (US PGPUB No. **20020116430**) and further in view of **Yeh et al.** (US Patent No. **6,848,043**).

With Regards to Claims 1, 21, 38, 53, 66, 67, Takahashi discloses a method

implemented in a device supporting a public-key cryptography application, the method comprising:

a first arithmetic circuit comprising a first plurality of arithmetic structures feeding back high order bits of a previously executed single arithmetic instruction of a processor instruction set in the public-key cryptography application, generated by the first arithmetic circuit, (see Takahashi paragraph [0009], lines 1-3: provides multi-function processor architecture capable of performing mathematic operations (implies arithmetic operations are part of system architecture or instruction set and pipeline processing initiated by a single instruction); paragraph [0017], lines 1-8: system for encrypting and decrypting data; includes an encryption/decryption engine operable to encrypt received data paragraph [0017], lines 8-14: one or more processor are operable to receive one or more operands and compute a result)

a second arithmetic circuit generating a first partial result of a currently executing single arithmetic instruction in the public-key cryptography application, wherein the currently executing single arithmetic instruction does not include an explicit source operand for specifying the high order bits, the first partial result representing the high order bits summed with low order bits of a result of a first number multiplied by a second number, the summing of the high order bits being performed during multiplication of the first number and the second number, the summing and at least a portion of the multiplication being performed in the second arithmetic circuit; (Takahashi paragraph [0038], lines

1-7: pipeline processing portion is coupled to operand storage portion and receives operands stored in operand storage portion; paragraph [0017], lines 1-19: pipeline storage processing (multiplication, summation) stage iteratively computing a running partial product using one or more received operands a predetermined number of times; post processing stage to receive final partial product and compute result; paragraph [0044], lines 1-14: receive bit stored in the current highest order position; first carry-save processor (a0 position); second carry-save processor (a1 position); third carry-save processor (a2); ...)

storing the first partial result; (see Takahashi paragraph [0041], lines 9-20: operand storage portion includes registers) and

using the stored first partial result in a subsequent computation in the public-key cryptography application. (see Takahashi paragraph [0017], lines 1-19: encryption/decryption engine operable to encrypt/decrypt received data; pipeline processing operable to receive operands; iteratively computing a running partial product using the one or more received operands)

Takahashi does not specifically disclose a partial result that comprises a high order portion of a result.

However, Chen discloses wherein the partial result comprises a high order portion of a result. (see Chen paragraph [0017], lines 14-19: hardware utilized to perform addition and multiplication; paragraph [0100], lines 12-18: output is high order bits, storing high order bits from adder; paragraph [0274], lines 16-19: carry feedback out

of high order position)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for a partial result comprises a high order portion of a result as taught by Chen. One of ordinary skill in the art would have been motivated to employ the teachings of Chen for the benefits achieved from the efficiency provided by a partitioning mechanism that enables the sharing of hardware. (Chen paragraph [0007], lines 11-13)

Takahashi-Chen does not specifically disclose feeding back a result (a number of bits) of a previously executed single arithmetic instruction.
Yeh discloses feeding back bits of a previously executed single arithmetic instruction (an operand transfer between two single instructions). (see Yeh col 5, ll 27-33: first instruction in a dependency chain accepts input from registers; second instruction accepts results of first instruction as it input operand)
The specification discloses an extended carry register to transfer operands between operations.

It would have been obvious to one of ordinary skill in the art to modify Takahashi-Chen for feeding back a result (a number of bits) of a previously executed single arithmetic instruction as taught by Yeh. One of ordinary skill in the art would have been motivated to employ the teachings of Stribaek for the benefits achieved from different approaches to improve the performance of arithmetic operations within computing systems.

With Regards to Claim 4, Takahashi discloses the method as recited in claim 1, further comprising feeding back the high order bits through a register to the second arithmetic circuit. (see Takahashi paragraph [0041], lines 9-20: operand storage portion includes registers; paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage portion; paragraph [0044], lines 1-14: carry-save processor (operations) feeds back highest order bit to next operation in pipeline)

Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

Yeh discloses feeding back the high order bits to the second arithmetic circuit as stated in Claim 1.

With Regards to Claim 5, Takahashi discloses the method as recited in claim 1 further comprising generating a second partial result of the currently executing single arithmetic instruction in the first arithmetic circuit, the second partial result representing the high order bits of the multiplication result of the first number multiplied by the second number. (see Takahashi paragraph [0041], lines 9-20: operand storage portion includes at least five registers; paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage portion; paragraph [0044], lines 1-14: carry-save feeds back highest order bit to next operation in pipeline)

Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

With Regards to Claim 6, Takahashi discloses the method as recited in claim 1 further comprising: discloses generating a second partial result of the currently executing single arithmetic instruction, the second partial result representing the high order bits of the multiplication result of the first number multiplied by the second number summed with the high order bits of the executed arithmetic instruction previously executed single arithmetic instruction. (see Takahashi paragraph [0041], lines 9-20: operand storage portion includes at least five registers; paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage portion; paragraph [0044], lines 1-14: carry-save feeds back highest order bit to next operation in pipeline)
Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.
Yeh discloses a previously executed single arithmetic instruction as stated in Claim 1.

With Regards to Claim 7, Takahashi discloses the method as recited in claim 6 further comprising supplying values generated in one or more most significant columns of the second arithmetic structures to one or more least significant columns of the first arithmetic structures while generating the first and second partial results. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion coupled to operand storage portion; iteratively computes running partial product a predetermined number of times using received operands; paragraph [0044], lines 1-14: receive bit stored in the current highest order position; first carry-save processor (a0 position); second carry-save processor (a1 position); third carry-save processor (a2); ...)

With Regards to Claim 8, Takahashi discloses the method as recited in claim 5 further comprising generating of the first and second partial result is in response to execution of a currently executing single arithmetic instruction. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion; iteratively computes a running partial product predetermined number of times using received operands)

With Regards to Claim 9, Takahashi discloses the method as recited in claim 6 further comprising generating of the first and second partial result is in response to execution of a currently executing single arithmetic instruction. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion; iteratively computes a running partial product predetermined number of times using received operands; (pipeline processing is a single instructions))

With Regards to Claim 10, Takahashi discloses the method as recited in claim 1 wherein at least one of the first and second pluralities of arithmetic structures comprises a plurality of carry save adder tree columns. (see Takahashi paragraph [0043], lines 6-13: pipeline processing stage includes carry-save processors (4 processors))

With Regards to Claim 19, Takahashi discloses the method as recited in claim 1 wherein feeding back high order bits of the currently executing arithmetic instruction from the first arithmetic circuit to the second arithmetic circuit for use with execution of a

subsequent single arithmetic instruction. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion; iteratively computes a running partial product predetermined number of times using received operands; (pipeline processing is a single instructions))

With Regards to Claim 22, Takahashi discloses the method as recited in claim 21, further comprising feeding back the high order bits through a register to the second arithmetic circuit. (see Takahashi paragraph [0041], lines 9-20: operand storage portion includes at least five registers; paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage portion)

With Regards to Claim 23, Takahashi discloses the method as recited in claim 21. the first arithmetic circuit generating a second partial result of the currently executing arithmetic instruction, the second partial result representing the high order bits of the multiplication result of the first number multiplied by the second number. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion coupled to operand storage portion; iteratively computes running partial product a predetermined number of times using received operands; paragraph [0044], lines 1-14: receive bit stored in the current highest order position; first carry-save processor (a0 position); second carry-save processor (a1 position); third carry-save processor (a2); ...)
Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

With Regards to Claim 24, Takahashi discloses the method as recited in claim 21. generating a second partial result of the currently executing arithmetic instruction, the second partial result representing the high order bits of the multiplication result of the first number multiplied by the second number summed with the high order bits of the previously executed arithmetic instruction and the third number. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion coupled to operand storage portion; iteratively computes running partial product a predetermined number of times using received operands; paragraph [0044], lines 1-14: receive bit stored in the current highest order position; first carry-save processor (a0 position); second carry-save processor (a1 position); third carry-save processor (a2); ...)
Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

With Regards to Claim 25, Takahashi discloses the method as recited in claim 24 wherein supplying values generated in one or more most significant columns of the second arithmetic structures to one or more least significant columns of the first arithmetic structures while generating the first and second partial results. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion coupled to operand storage portion; iteratively computes running partial product a predetermined number of times using received operands; paragraph [0044], lines 1-14: receive bit stored in the current highest order position; first carry-save processor (a0 position); second carry-

save processor (a1 position); third carry-save processor (a2); ...)

Takahashi does not explicitly disclose a partial result comprises a least significant portion of a result.

However, Chen discloses wherein the partial result comprises a least significant portion of a result. (see Chen paragraph [0017], lines 14-19: hardware utilized to perform addition and multiplication; paragraph [0097], lines 1-5: low order bits from multiple supplied to adder; paragraph [0097], lines 14-17: output of adder is supplied to multiplier which multiplies two numbers)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for a partial result comprises a least significant portion of a result as taught by Chen. One of ordinary skill in the art would have been motivated to employ the teachings of Chen for the benefits achieved from the efficiency provided by a partitioning mechanism that enables the sharing of hardware. (Chen paragraph [0007], lines 11-13)

With Regards to Claim 26, Takahashi discloses the method as recited in claim 23 wherein generating of the first and second partial result is in response to execution of a single arithmetic instruction. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage; pipeline processing iteratively computes a running partial product a predetermined number of times using received operands)

With Regards to Claim 36, Takahashi discloses the method as recited in claim 21

further comprising feeding back high order bits of the currently executing arithmetic instruction from the first arithmetic circuit to the second arithmetic circuit for use with execution of a subsequent single arithmetic instruction. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion coupled to operand storage portion; iteratively computes running partial product a predetermined number of times using received operands; paragraph [0044], lines 1-14: receive bit stored in the current highest order position; first carry-save processor (a0 position); second carry-save processor (a1 position); third carry-save processor (a2); ...)
Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

With Regards to Claim 39, Takahashi discloses the processor as recited in claim 38. the first arithmetic structures are configured to generate a second partial result of the arithmetic instruction, the second partial result representing the high order bits of the arithmetic operation. (see Takahashi paragraph [0044], lines 1-14: receive bit stored in the current highest order position; first carry-save processor (a0 position); second carry-save processor (a1 position); third carry-save processor (a2); ...)
Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

With Regards to Claim 40, Takahashi discloses the processor as recited in claim 39. the second arithmetic structures are further configured to supply values generated in

one or more most significant columns of the second arithmetic structures to one or more least significant columns of the first arithmetic structures while generating the first and second partial results. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion coupled to operand storage portion; iteratively computes running partial product a predetermined number of times using received operands; paragraph [0044], lines 1-14: receive bit stored in the current highest order position; first carry-save processor (a0 position); second carry-save processor (a1 position); third carry-save processor (a2); ...)

With Regards to Claim 41, Takahashi discloses the processor as recited in claim 39, wherein the first and second arithmetic structures are configured to generate of the first and second partial results in response to execution of a single arithmetic instruction. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion coupled to operand storage portion; iteratively computes running partial product a predetermined number of times using received operands)

With Regards to Claim 42, Takahashi discloses the processor as recited in claim 38, further comprising a register coupled to the first and second arithmetic structures to supply the high order bits to the second arithmetic structures. (see Takahashi paragraph [0041], lines 9-20: operand storage portion includes at least five registers; paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage portion)

Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

With Regards to Claim 48, Takahashi discloses the processor as recited in claim 38. at least one of the first and second pluralities of arithmetic structures comprises a plurality of carry save adder tree columns. (see Takahashi paragraph [0043], lines 6-13: pipeline processing stage includes multiple carry-save processors (4 processors))

With Regards to Claim 52, Takahashi discloses the processor as recited in claim 38, wherein the processor is a general purpose processor. (see Takahashi paragraph [0003], lines 1-5: processor for performing operations)

With Regards to Claim 54, Takahashi discloses the processor as recited in claim 53, wherein the first arithmetic structures are further configured to generate a second partial result of the arithmetic instruction, the second partial result representing the high order bits of the arithmetic operation. (Takahashi paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage portion; paragraph [0044], lines 1-14: receive bit stored in the current highest order position; first carry-save (a0 position); second carry-save (a1 position,...))
Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

With Regards to Claim 55, Takahashi discloses the processor as recited in claim 54, wherein the second arithmetic structures are further configured to generate values in one or more most significant columns and to supply them to one or more least significant columns of the first arithmetic structures while generating the first and second partial results. (Takahashi paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage portion; paragraph [0044], lines 1-14: receive bit stored in the current highest order position; first carry-save processor (a0 position); second carry-save processor (a1 position); third carry-save processor (a2); ...)
Takahashi does not explicitly disclose a partial result comprises a least significant portion of a result.
However, Chen discloses wherein the partial result comprises a least significant portion of a result. (see Chen paragraph [0017], lines 14-19: hardware utilized to perform addition and multiplication; paragraph [0097], lines 1-5: low order bits from multiple supplied to adder; paragraph [0097], lines 14-17: output of adder is supplied to multiplier which multiplies two numbers)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for a partial result comprises a least significant portion of a result as taught by Chen. One of ordinary skill in the art would have been motivated to employ the teachings of Chen for the benefits achieved from the efficiency provided by a partitioning mechanism that enables the sharing of hardware. (Chen paragraph [0007], lines 11-13)

With Regards to Claim 56, Takahashi discloses the processor as recited in claim 54,

wherein the first arithmetic structures are configured to generate of the first and second partial result in response to execution of a single arithmetic instruction. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage portion; pipeline computes a running partial product a predetermined number of times; at least a portion of the one or more received operands in each iteration; (implies a single arithmetic execution initiate an iteration of arithmetic operations))

With Regards to Claim 57, Takahashi discloses the processor as recited in claim 53, a further comprising a register coupled to the first and second arithmetic structures to supply the high order bits to the second arithmetic structures. (see Takahashi paragraph [0041], lines 9-20: operand storage portion includes at least five registers; paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage portion) Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

With Regards to Claim 58, Takahashi discloses the processor as recited in claim 53, further comprising an adder circuit configured to receive the first partial result and to generate a non redundant representation of the first partial result and a carry out value. (see Takahashi paragraph [0012], lines 1-4: performing a carry-save add of at least a first operand)

With Regards to Claim 59, Takahashi discloses the processor as recited in claim 58

wherein the adder circuit is further configured to feed the carry out value back to itself as an input. (see Takahashi paragraph [0045], lines 1-11: carry-save processor are in a ring configuration; output of fourth carry-save processor is input to first carry-save processor; (carry out value eventually input to itself))

With Regards to Claim 60, Takahashi discloses the processor as recited in claim 58, wherein the adder circuit is further configured to feed the carry out value back to the second arithmetic structures. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage portion; iteratively computes a partial product a predetermined number of times using a portion of one or more received operands)

With Regards to Claim 62, Takahashi discloses the processor as recited in claim 53, wherein at least one of the first and second arithmetic structures comprises carry save adder tree columns. (see Takahashi paragraph [0012], lines 1-5: carry-save processor for performing carry-save add of at least a first operand, second operand, third operand)

8. Claims **2, 3, 15 - 18, 27 - 29, 35, 43 - 46** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Takahashi** in view of **Chen** and further in view of **Yeh** and **Vanstone et al.** (US Patent No. **6,735,611**).

With Regards to Claim 2, Takahashi discloses the method as recited in claim 1,

wherein the high order bits are fed back. (see Takahashi paragraph [0038, lines 1-7: : pipeline processing; iteratively computes a partial product a predetermined number of times using received operands; paragraph [0012], lines 1-5; paragraph [0043], lines 6-13: pipeline processing stage includes a carry-save processor; (carry-save implies high order bits are carried forward to the next operand); paragraph [0044], lines 1-14: carry-save operation utilizes current high order bit position)

Takahashi does not specifically disclose redundant number representation.

However, Vanstone discloses wherein the result is in redundant number representation. (see Vanstone col 9, lines 59-64: when multiplying, provides a redundant representation of partial product)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for a result in redundant number representation as taught by Vanstone. One of ordinary skill in the art would have been motivated to employ the teachings of Lasher or the benefits achieved from arithmetic processors supporting EC and RSA cryptography with no penalty in performance and cost. (see Vanstone col. 1, lines 40-43)

Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

With Regards to Claim 3, Takahashi discloses the method as recited in claim 2 includes sum and carry bits.

Takahashi does not specifically disclose redundant number representation.

However, Vanstone discloses wherein the result is in redundant number representation.

(see Vanstone col 9, lines 59-64: when multiplying, provides a redundant representation of partial product)

Motivation for Vanstone to disclose redundant number representation is as stated in Claim 2 above.

With Regards to Claim 15, Takahashi discloses the method as recited in claim 1 wherein the first partial result. (see Takahashi paragraph [0038, lines 1-7: pipeline processing; iteratively computes a partial product a predetermined number of times using received operands)

Takahashi does not specifically disclose redundant number representation.

However, Vanstone discloses wherein the result is in redundant number representation. (see Vanstone col 9, lines 59-64: when multiplying, provides a redundant representation of partial product)

Motivation for Vanstone to disclose redundant number representation is as stated in Claim 2 above.

With Regards to Claim 16, Takahashi discloses the method as recited in claim 15 further comprising supplying the first partial result to an adder circuit to generate the first partial result and a carry out value. (see Takahashi paragraph [0038, lines 1-7: pipeline processing; iteratively computes a partial product a predetermined number of times using received operands)

Takahashi does not specifically disclose redundant number representation.

However, Vanstone discloses wherein the result is a non redundant representation. (see Vanstone col 9, lines 59-64: when multiplying, provides a redundant representation of partial product)

Motivation for Vanstone to disclose redundant number representation is as stated in Claim 2 above.

With Regards to Claim 17, Takahashi discloses the method as recited in claim 16 further comprising feeding back the carry out value to the adder circuit. (see Takahashi paragraph [0038, lines 1-7: pipeline processing; iteratively computes a partial product a predetermined number of times using received operands; paragraph [0012], lines 1-5; paragraph [0043], lines 6-13: pipeline processing stage includes a carry-save processor; (carry-save implies high order bits are carried forwards high order bits using operand))

With Regards to Claim 18, Takahashi discloses the method as recited in claim 16 further comprising feeding back the carry out value to the second arithmetic circuit. (see Takahashi paragraph [0038, lines 1-7: pipeline processing; iteratively computes a partial product a predetermined number of times using received operands; paragraph [0012], lines 1-5; paragraph [0043], lines 6-13: pipeline processing stage includes a carry-save processor; (carry-save implies high order bits are carried forwards high order bits using operand))

With Regards to Claim 27, Takahashi discloses the method as recited in claim 21 further comprising supplying the first partial result to an adder circuit to generate a non redundant representation of the first partial result and a carry out value. (see Takahashi paragraph [0017], lines 11-19: each processor includes operand storage, pipeline processing portion, post-processing stage; pipeline processing iteratively computing a running partial product using one or more received operands)
Takahashi does not specifically disclose redundant number representation.
However, Vanstone discloses wherein the result is a non redundant number representation. (see Vanstone col 9, lines 59-64: when multiplying, provides a redundant representation of partial product)
Motivation for Vanstone to disclose redundant number representation is as stated in Claim 2 above.

With Regards to Claim 28, Takahashi discloses the method as recited in claim 27 further comprising feeding back the carry out value to the adder circuit. (see Takahashi paragraph [0038, lines 1-7: pipeline processing; iteratively computes a partial product a predetermined number of times using received operands; paragraph [0012], lines 1-5; paragraph [0043], lines 6-13: pipeline processing stage includes a carry-save processor; (carry-save implies high order bits are carried forwards high order bits using operand))

With Regards to Claim 29, Takahashi discloses the method as recited in claim 27

further comprising feeding back the carry out value to the second arithmetic structures. (see Takahashi paragraph [0017], lines 11-19: each processor includes operand storage, pipeline processing portion, post-processing stage; pipeline processing iteratively computing a running partial product using one or more received operands)

With Regards to Claim 35, Takahashi discloses the method as recited in claim 21 wherein the high order bits. (see Takahashi paragraph [0038, lines 1-7: pipeline processing; iteratively computes a partial product a predetermined number of times using received operands; paragraph [0012], lines 1-5; paragraph [0043], lines 6-13: pipeline processing stage includes a carry-save processor; (carry-save implies high order bits are carried forwards high order bits using operand))
Vanstone discloses the result is in redundant number representation as stated in Claim 2 above.
Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

With Regards to Claim 43, Takahashi discloses the processor as recited in claim 38, wherein the first partial result. (see Takahashi paragraph [0038, lines 1-7: pipeline processing; iteratively computes a partial product a predetermined number of times using received operands)
Takahashi does not specifically disclose redundant number representation.
Vanstone discloses the result is in redundant number representation as stated in Claim

2 above.

With Regards to Claim 44, Takahashi discloses the processor as recited in claim 43 further comprising an adder circuit configured to receive the first partial result and to generate a non redundant representation of the first partial result and a carry out value. (see Takahashi paragraph [0038, lines 1-7: pipeline processing; iteratively computes a partial product a predetermined number of times using received operands; paragraph [0012], lines 1-5; paragraph [0043], lines 6-13: pipeline processing stage includes a carry-save processor; (carry-save implies high order bits are carried forwards high order bits using operand))

With Regards to Claim 45, Takahashi discloses the processor as recited in claim 44 further comprising an adder circuit configured to feed the carry out value back to itself as an input. (see Takahashi paragraph [0045], lines 1-11: carry-save processor are in a ring configuration; output of fourth carry-save processor is input to first carry-save processor; (carry out value eventually input to itself))

With Regards to Claim 46, Takahashi discloses the processor as recited in claim 44 further comprising an adder circuit configured to feed the carry out value back to the second arithmetic structures. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage portion; iteratively computes a partial product a predetermined number of times using a portion of one or more received

operands)

9. Claims **11, 20, 30, 31, 37, 47, 61** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Takahashi** in view of **Chen** and further in view of **Yeh** and **Stribaek et al.** (US Patent No. **7,181,484**).

With Regards to Claim 11, Takahashi discloses the method as recited in claim 1, further comprising at least one of the first and second pluralities of arithmetic structures. (see Takahashi paragraph [0017], lines 11-19: each processor includes operand storage, pipeline processing portion, post-processing stage; pipeline processing iteratively computing a running partial product using one or more received operands) Takahashi does not specifically disclose whereby a plurality of Wallace tree columns. However, Stribaek discloses wherein further comprises a plurality of Wallace tree columns. (see Stribaek col. 9, lines 10-24; col. 9, lines 37-39: Wallace tree)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for usage of Wallace tree multiplication as taught by Stribaek. One of ordinary skill in the art would have been motivated to employ the teachings of Stribaek in order to enable the capability for extended precision in arithmetic calculations due to extensive and increasing usage of public key cryptography. (see Stribaek col. 1, lines 61-67)

With Regards to Claim 20, Takahashi discloses the method as recited claim 1, further comprising storing the high order bits.

Takahashi does not specifically disclose whereby an extended carry register. However, Stribaek discloses wherein an extended carry register. (see Stribaek col. 9, lines 10-24; col. 9, lines 37-39: Wallace tree; col. 5, lines 41-45: extended carry operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for usage of extended carry operations as taught by Stribaek. One of ordinary skill in the art would have been motivated to employ the teachings of Stribaek in order to enable the capability for extended precision in arithmetic calculations due to extensive and increasing usage of public key cryptography. (see Stribaek col. 1, lines 61-67) Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

With Regards to Claim 30, Takahashi discloses the method as recited in claim 21, wherein at least one of the first and second pluralities of arithmetic structures. Takahashi does not specifically disclose a plurality of Wallace tree columns. However, Stribaek discloses wherein further comprises a plurality of Wallace tree columns. (see Stribaek col. 9, lines 10-24; col. 9, lines 37-39: Wallace tree; col. 2, line 66 - col. 3, line 6: public key cryptographic calculations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for usage of Wallace tree multiplication as taught by Stribaek. One of ordinary skill in the art would have been motivated to employ the teachings of Stribaek in order to enable the capability for extended precision in arithmetic calculations due to extensive

and increasing usage of public key cryptography. (see Stribaek col. 1, lines 61-67)

With Regards to Claim 31, Takahashi discloses the method as recited in claim 21, wherein at least one of the first and second pluralities of arithmetic structures.
Takahashi does not specifically disclose carry save adder tree columns.
However, Stribaek discloses wherein further comprises a plurality of adder tree columns. (see Stribaek col. 9, lines 10-24; col. 9, lines 37-39: Wallace tree; col. 5, lines 41-45: extended carry operations; col. 7, lines 31-37; col. 9, lines 10-14: carry-save adder; col. 2, line 66 - col. 3, line 6: public key cryptographic calculations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for usage of Wallace tree multiplication as taught by Stribaek. One of ordinary skill in the art would have been motivated to employ the teachings of Stribaek in order to enable the capability for extended precision in arithmetic calculations due to extensive and increasing usage of public key cryptography. (see Stribaek col. 1, lines 61-67)

With Regards to Claim 37, Takahashi discloses the method as recited in claim 21, further comprising storing the high order bits.
Takahashi does not specifically disclose an extended carry register.
However, Stribaek discloses wherein an extended carry register. (see Stribaek col. 9, lines 10-24; col. 9, lines 37-39: Wallace tree; col. 5, lines 41-45: extended carry operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi

for an extended carry register as taught by Stribaek. One of ordinary skill in the art would have been motivated to employ the teachings of Stribaek in order to enable the capability for extended precision in arithmetic calculations due to extensive and increasing usage of public key cryptography. (see Stribaek col. 1, lines 61-67) Chen discloses a partial result representing the high order bits of a result as stated in Claim 1.

With Regards to Claim 47, Takahashi discloses the processor as recited in claim 38, wherein at least one of the first and second pluralities of the arithmetic structures. Takahashi does not specifically disclose whereby a plurality of Wallace tree columns. However, Stribaek discloses wherein further comprises a plurality of Wallace tree columns. (see Stribaek col. 9, lines 10-24; col. 9, lines 37-39: Wallace tree)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for a plurality of Wallace tree columns as taught by Stribaek. One of ordinary skill in the art would have been motivated to employ the teachings of Stribaek in order to enable the capability for extended precision in arithmetic calculations due to extensive and increasing usage of public key cryptography. (see Stribaek col. 1, lines 61-67)

With Regards to Claim 61, Takahashi discloses the processor as recited in claim 53, wherein at least one of the first and second arithmetic structures.
Huppenthal does not specifically disclose Wallace tree columns.
However, Stribaek discloses wherein further comprises a Wallace tree column. (see

Stribaek col. 9, lines 10-24; col. 9, lines 37-39: Wallace tree)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for Wallace tree columns as taught by Stribaek. One of ordinary skill in the art would have been motivated to employ the teachings of Stribaek in order to enable the capability for extended precision in arithmetic calculations due to extensive and increasing usage of public key cryptography. (see Stribaek col. 1, lines 61-67)

10. Claims **12 - 14, 32 - 34, 49 - 51, 63 - 65** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Takahashi** in view of **Chen** and further in view of **Yeh** and **Chen et al.** (US Patent No. **6,687,725** referred as "Chen2").

With Regards to Claim 12, Takahashi discloses the method as recited in claim 1, wherein at least one of the first and second pluralities of arithmetic structures is usable to perform integer multiplication.

Takahashi does not specifically disclose XOR operations.

However, Chen2 discloses wherein to perform XOR multiplication. (see Chen2 col. 4, line 64 - col. 5, line 2; col. 15, lines 29-31: XOR operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi to perform XOR multiplication as taught by Chen2 One of ordinary skill in the art would have been motivated to employ the teachings of Chen2 in order to provide an arithmetic circuit which can perform all arithmetic operations in the finite field, including addition, multiplication, division, exponentiation and inverse multiplication. (see Chen2 col. 3,

lines 17-21)

With Regards to Claim 13, Takahashi discloses the method as recited in claim 12, further comprising a logical circuit in at least one of the first and second arithmetic circuits supplying a variable value for integer multiplication mode that varies according to inputs supplied to the logical circuit if in integer multiplication mode, to thereby ensure a result unaffected by carry logic performing carries in integer multiplication mode. (see Takahashi paragraph [0038], lines 1-7: pipeline processing portion is coupled to operand storage; pipeline processing iteratively computes a running partial product a predetermined number of times using received operands; paragraph [0016]: arithmetic operations performed with integers)

Takahashi does not specifically disclose XOR operations.

However, Chen2 discloses wherein supplying a fixed value if in XOR multiplication mode and to thereby ensure a result is determined in XOR multiplication. (see Chen2 col. 4, line 64 - col. 5, line 2; col. 15, lines 29-31: XOR operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi to support XOR operations as taught by Chen2. One of ordinary skill in the art would have been motivated to employ the teachings of Chen2 in order to provide an arithmetic circuit which can perform all arithmetic operations in the finite field, including addition, multiplication, division, exponentiation and inverse multiplication. (see Chen2 col. 3, lines 17-21)

With Regards to Claim 14, Takahashi discloses the method as recited in claim 13 wherein the logical circuit operates as a majority circuit in integer multiplication mode.
Takahashi does not specifically disclose XOR operations.
However, Chen2discloses wherein outputs a zero in the XOR multiplication mode. (see Chen2 col. 4, line 64 - col. 5, line 2; col. 15, lines 29-31: XOR operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi to support XOR operations as taught by Chen2. One of ordinary skill in the art would have been motivated to employ the teachings of Chen2 in order to provide an arithmetic circuit which can perform all arithmetic operations in the finite field, including addition, multiplication, division, exponentiation and inverse multiplication. (see Chen2 col. 3, lines 17-21)

With Regards to Claim 32, Takahashi discloses the method as recited in claim 21, wherein at least one of the first and second pluralities of arithmetic structures is usable to perform integer multiplication.
Takahashi does not specifically disclose XOR operations.
However, Chen2 discloses wherein perform both integer and XOR multiplication. (see Chen2 col. 4, line 64 - col. 5, line 2; col. 15, lines 29-31: XOR operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi to perform XOR multiplication as taught by Chen2. One of ordinary skill in the art would have been motivated to employ the teachings of Chen2 in order to provide an arithmetic circuit which can perform all arithmetic operations in the finite field, including addition,

multiplication, division, exponentiation and inverse multiplication. (see Chen2 col. 3, lines 17-21)

With Regards to Claim 33, Takahashi discloses the method as recited in claim 32 wherein a logic circuit in at least one of the first and second pluralities of arithmetic structures supplying a variable value that varies according to inputs supplied to the logical circuit if in integer multiplication mode, to thereby ensure a result unaffected by carry logic performing carries in integer multiplication mode.
Takahashi does not specifically disclose XOR operations.
However, Chen2 discloses wherein supplying a fixed value if in XOR multiplication mode. (see Chen2 col. 4, line 64 - col. 5, line 2; col. 15, lines 29-31: XOR operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for XOR operations as taught by Chen2. One of ordinary skill in the art would have been motivated to employ the teachings of Chen2 in order to provide an arithmetic circuit which can perform all arithmetic operations in the finite field, including addition, multiplication, division, exponentiation and inverse multiplication. (see Chen2 col. 3, lines 17-21)

With Regards to Claim 34, Takahashi discloses the method as recited in claim 33 wherein the logic circuit operates as a majority circuit in integer multiplication mode.
Takahashi does not specifically disclose XOR operations.
However, Chen2discloses wherein outputs a zero in the XOR multiplication mode. (see

Chen2 col. 4, line 64 - col. 5, line 2; col. 15, lines 29-31: XOR operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for XOR operations as taught by Chen2. One of ordinary skill in the art would have been motivated to employ the teachings of Chen2 in order to provide an arithmetic circuit which can perform all arithmetic operations in the finite field, including addition, multiplication, division, exponentiation and inverse multiplication. (see Chen2 col. 3, lines 17-21)

With Regards to Claim 49, Takahashi discloses the processor as recited in claim 38, wherein at least one of the first and second pluralities of arithmetic structures is configured to selectively perform one of integer multiplication according to a control signal.

Takahashi does not specifically disclose XOR operations.

However, Chen2 discloses wherein to selectively perform XOR multiplication. (see Chen2 col. 4, line 64 - col. 5, line 2; col. 15, lines 29-31: XOR operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for XOR operations as taught by Chen2. One of ordinary skill in the art would have been motivated to employ the teachings of Chen2 in order to provide an arithmetic circuit which can perform all arithmetic operations in the finite field, including addition, multiplication, division, exponentiation and inverse multiplication. (see Chen2 col. 3, lines 17-21)

With Regards to Claim 50, Takahashi discloses the processor as recited in claim 49 further comprising a plurality of logic circuits in the first and second pluralities of arithmetic structures, each logic circuit responsive to the control signal to supply a variable output value in integer multiplication mode, the variable output value varying according to values of inputs supplied to the logic circuit, to thereby ensure a result unaffected by carry logic generating carries in integer multiplication mode.
Takahashi does not specifically disclose XOR multiplication mode.
However, Chen2 discloses to supply a fixed output value in XOR multiplication mode and ensure a result is determined in XOR multiplication mode a fixed output value in XOR multiplication mode. (see Chen2 col. 4, line 64 - col. 5, line 2; col. 15, lines 29-31: XOR operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for XOR multiplication mode as taught by Chen2. One of ordinary skill in the art would have been motivated to employ the teachings of Chen2 in order to provide an arithmetic circuit which can perform all arithmetic operations in the finite field, including addition, multiplication, division, exponentiation and inverse multiplication. (see Chen2 col. 3, lines 17-21)

With Regards to Claim 51, Takahashi discloses the processor as recited in claim 50, wherein the logical circuit is configured to operate as a majority circuit in integer multiplication mode.
Takahashi does not specifically disclose XOR operations.

However, Chen2 discloses wherein to output a zero in XOR multiplication mode. (see Chen2 col. 4, line 64 - col. 5, line 2; col. 15, lines 29-31: XOR operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi to output a zero in XOR multiplication mode as taught by Chen2. One of ordinary skill in the art would have been motivated to employ the teachings of Chen2 in order to provide an arithmetic circuit which can perform all arithmetic operations in the finite field, including addition, multiplication, division, exponentiation and inverse multiplication. (see Chen2 col. 3, lines 17-21)

With Regards to Claim 63, Takahashi discloses the processor as recited in claim 53, wherein the arithmetic structures are configured to selectively perform one of integer multiplication according to a control signal.

Takahashi does not specifically disclose XOR multiplication.

However, Chen2 discloses wherein to perform XOR multiplication. (see Chen2 col. 4, line 64 - col. 5, line 2; col. 15, lines 29-31: XOR operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi for XOR multiplication as taught by Chen2. One of ordinary skill in the art would have been motivated to employ the teachings of Chen2 in order to provide an arithmetic circuit which can perform all arithmetic operations in the finite field, including addition, multiplication, division, exponentiation and inverse multiplication. (see Chen2 col. 3, lines 17-21)

With Regards to Claim 64, Takahashi discloses the processor as recited in claim 63. Hinds discloses a plurality of logic circuits in at least one of the first and second pluralities of arithmetic structures, each logic circuit responsive to the control signal to supply a variable output value in integer multiplication mode, the variable output value varying according to values of inputs supplied to the logic circuit, to thereby ensure a result is unaffected by carry logic generating carries in integer multiplication mode as stated in Claim 1 above.

Huppenthal does not specifically disclose XOR operations.

However, Chen2 discloses wherein to supply a fixed output value in XOR multiplication mode and to thereby ensure a result is determined in XOR multiplication mode. (see Chen2 col. 4, line 64 - col. 5, line 2; col. 15, lines 29-31: XOR operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi to support XOR operations as taught by Chen2. One of ordinary skill in the art would have been motivated to employ the teachings of Chen2 in order to provide an arithmetic circuit which can perform all arithmetic operations in the finite field, including addition, multiplication, division, exponentiation and inverse multiplication. (see Chen2 col. 3, lines 17-21)

With Regards to Claim 65, Takahashi discloses the processor as recited in claim 64, wherein the logical circuit is configured to operate as a majority circuit in integer multiplication mode.

Takahashi does not specifically disclose XOR operations.

However, Chen2 discloses wherein to output a zero in the XOR multiplication mode. (see Chen2 col. 4, line 64 - col. 5, line 2; col. 15, lines 29-31: XOR operations)

It would have been obvious to one of ordinary skill in the art to modify Takahashi and to output a zero in the XOR multiplication mode as taught by Chen2. One of ordinary skill in the art would have been motivated to employ the teachings of Chen2 in order to provide an arithmetic circuit which can perform all arithmetic operations in the finite field, including addition, multiplication, division, exponentiation and inverse multiplication. (see Chen2 col. 3, lines 17-21)

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Carlton V. Johnson whose telephone number is 571-270-1032. The examiner can normally be reached on Monday thru Friday , 8:00 - 5:00PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nasser Moazzami can be reached on 571-272-4195. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/David García Cervetti/
Primary Examiner, Art Unit 2436

Carlton V. Johnson
Examiner
Art Unit 2436

CVJ
September 12, 2011